

Документ подписан электронной подписью
Информация о владельце:
ФИО: Косенок Сергей Михайлович
Должность: ректор
Дата подписания: 20.06.2024 11:24:55
Уникальный программный ключ:
e3a68f3eaa1a67674b541d99819c41d661d1836

Оценочные материалы для промежуточной аттестации по дисциплине

Инструменты и технологии web-разработки

Код, направление	09.04.04 ПРОГРАММНАЯ ИНЖЕНЕРИЯ
подготовки	
Направленность (профиль)	Разработка и интеграция информационных систем и сервисов
Форма обучения	Очная
Кафедра-разработчик	Автоматики и компьютерных систем
Выпускающая кафедра	Автоматики и компьютерных систем

Этап: проведение текущего контроля успеваемости по дисциплине

Раздел 1. Основы web-технологий

Тема: Верстка HTML-страниц. Задание на лабораторную работу:

С помощью инструментов разработчика браузера Google Chrome проанализировать структуру HTTP-запроса и ответа, отправляемого браузером для загрузки страницы. Выписать значения полей запроса и ответа в таблицу. Привести скриншот в отчете.

Согласно своему варианту предметной области создать набор веб-страниц. Приветствуются инициативные темы.

Главная страница (index.html) имеет три фрейма – «Категории», «Объекты», «Описания».

Фрейм «Категории» должен содержать перечень ссылок на категории объектов предметной области. Например, для предметной области «Реактивные самолеты»: «Гражданские», «Учебно-тренировочные», «Сверхзвуковые» т.д., всего не менее 3-х категорий.

Фрейм «Объекты» должен содержать общее описание выбранной категории и перечень ссылок на объекты этой категории, включая их уменьшенные изображения, не менее 3-х объектов для каждой категории.

Фрейм «Описания» должен содержать описание выбранного объекта – его характеристики, свойства, изображения и т.д.

Таким образом, проект должен содержать следующие файлы (имена файлов могут быть изменены):

index.html - фреймовая структура;

categories.html - перечень ссылок на категории и ссылка «главная страница»;

category1.html .. category3.html – описания категорий и ссылки на объекты категории;

object1-1.html, object1-2.html .. object3-3.html – описания объектов.

При открытии главной страницы в фрейме «Категории» должен отображаться список категорий. Фрейм «Объекты» должен содержать информацию «О проекте», фрейм «Описания» должен содержать описание предметной области в целом. Перечень категорий должен содержать ссылку «Главная страница», при выборе которой страница принимает первоначальный вид.

При выполнении задания необходимо использовать все элементы HTML, описанные в данной лабораторной работе!

Тема: Применение CSS-стилей. Задание на лабораторную работу:

1. Самостоятельно изучить свойства CSS.
2. Создать css-файл и подключить его ко всем html-документам, разработанным в лабораторной работе 1.
3. Исходя из соображений дизайна страниц разработать стили с использованием HTML-селекторов, селекторов класса и ID-селекторов. При этом следует учесть, что HTML-селекторы будут распространяться на все страницы, в которых встречаются соответствующие тэги (элементы). Все страницы одного типа, например, описания объектов, должны быть унифицированы и использовать одни и те же стили CSS.

Разработать веб-страницу, содержащую форму с полями для описания объекта предметной области в соответствии с вариантом задания на лабораторную работу №1.

В форме необходимо использовать поля текстового, многострочного текстового типов, радиокнопки, чекбокс, выпадающий список, список с множественным выбором.

В веб-странице реализовать сценарий JavaScript, выполняющий проверку правильности вводимых в форму данных и вывод их в виде описания объекта на этой же странице.

Форма должна иметь оригинальный дизайн, заданный при помощи CSS.

Веб-страница с формой должна отображаться в фрейме «описания» по ссылке из списка категорий.

Тема: Язык программирования JavaScript. Задание на лабораторную работу:

На страницу с перечнем категорий добавить форму для удаления и добавления категорий. Предусмотреть поля для ввода наименования категории и имени файла, который будет открываться при выборе категории. Пример меню категорий, реализованного в виде списка:

Список:

- [Категория 1](#)
- [Категория 2](#)
- [Категория 3](#)

Добавление категории:

Имя категории:

Ссылка:

Список:

- [Категория 1](#)
- [Категория 2](#)
- [Категория 3](#)
- [Категория 4](#)

Добавление категории:

Имя категории:

Ссылка:

На страницы с перечнем объектов добавить форму для удаления и добавления объектов. Предусмотреть поля для ввода наименования объекта, имени файла, который будет открываться при выборе объекта. Пример меню объектов, реализованного в виде таблицы с добавлением ячеек в строку:

Объекты категории:

Объект 1	Объект 2	Объект 3
----------	----------	----------

Объекты категории:

Объект 1	Объект 2	Объект 3	Объект 4
			<input type="checkbox"/>

Имя объекта:

Ссылка:

Имя объекта:

Ссылка:

Примечание: при реализации для добавления категорий использовать список, для добавления объектов – таблицу, или наоборот. Список удобнее использовать при вертикальном расположении элементов меню.

На страницу с перечнем категорий добавить форму для выбора «темы» сайта. Для этого можно использовать подключение к странице различных css-файлов с заранее разработанными стилями при помощи элемента <link>, который находится в заголовке:

Таблица:

Столбец1	Столбец2	Столбец3
Столбец1	Столбец2	Столбец3

Стиль таблицы:

Таблица:

Столбец1	Столбец2	Столбец3
Столбец1	Столбец2	Столбец3

Стиль таблицы:

Выбор темы должен сопровождаться изменением дизайна отображения страниц во всех трех фреймах. Для доступа к содержимому других фреймов см. п. 4.3.8.

Раздел 2. Динамические страницы и серверное программирование

Тема: Основы реляционных баз данных и языка SQL. Задание на лабораторную работу:

1. Написать сценарий (набор SQL-команд) для создания базы данных «Предприятие».
2. Написать сценарий, функционально аналогичный сценарию для создания и заполнения данными БД «Предприятие» согласно своему варианту задания.
3. Построить диаграмму БД «Предприятие».
4. Произвести экспорт схемы (таблиц) и данных БД «Предприятие».
6. Построить диаграмму БД согласно индивидуальному заданию.
7. Произвести экспорт схемы (таблиц) и данных БД согласно индивидуальному заданию.

Тема: Язык программирования PHP. Задание на лабораторную работу:

Перенести все файлы сайта, созданного в предыдущих работах, на сервер в указанную преподавателем папку. Проверить работу сайта.

Обработку формы, разработанной по заданию в лабораторной работе №3, реализовать на стороне сервера в виде сценария PHP. При этом необходимо выполнить проверку правильности ввода данных на стороне клиента в сценарии JavaScript.

Тема: Управление контролем версий Git и командная работа над проектом. Задание на лабораторную работу:

Создайте репозиторий на GitHub, клонируйте его на локальную машину. Перейдите в папку, появившуюся после клонирования командой `cd имя-папки`. Переходить в папку репозитория нужно после каждого клонирования.

При помощи текстового редактора в папке репозитория на локальном компьютере создайте файл `README.md` и добавьте в него описание в формате Markdown. Можете скопировать разметку из подсказки и вставить её в файл `README.md`. У файла `README.md` не должно быть никаких других расширений. Например, имя файла `README.md.txt` будет считаться ошибкой.

Сохраните файл. При помощи команды `git status` проверьте, видит ли Git новый файл с названием `README.md`.

Сделайте коммит изменений;

Запушьте все изменения на GitHub;

Убедитесь, что через веб-интерфейс GitHub можно посмотреть список файлов, список коммитов, изменения, сделанные каждым коммитом.

В качестве результата пришлите

Ссылку на ваш репозиторий на GitHub;

Ссылку на последний коммит. Для этого на сайте GitHub нажмите на количество коммитов (под кнопкой Code), нажмите на последний коммит и скопируйте адрес из адресной строки.

Тема: Технологии развертывания веб-приложений и сервисов. Задание на лабораторную работу:

1. Зарегистрироваться на Heroku (<https://id.heroku.com/login>) и войти со своей учетной записью
2. Создать приложение на Heroku
3. Запустить терминал и установить интерпретатор команд Heroku - Heroku CLI (Command Line Interface)
4. В терминале выполнить команду `heroku login`. Затем нажать любую клавишу. Откроется окно браузера, в котором нужно пройти аутентификацию, после чего перейти обратно в терминал. Окно браузера можно закрыть.
5. Перейти в папку проекта и инициализировать репозиторий для Heroku
6. В корневой папке проекта создать файл с именем `Procfile`, который содержит одну строку: `web: vendor/bin/heroku-php-apache2 public/`
7. Локальную версию синхронизировать с репозиторием Heroku. Для этого в терминале в папке проекта необходимо выполнить команды: `git add . git commit -am "any comment" git push heroku HEAD:master`
8. Сконфигурировать переменную окружения `baseURL` в Heroku (аналог параметра в `.env` файле). Для этого на странице приложения открыть вкладку Settings и добавить параметр `baseURL` со значением, содержащим URL приложения `<имя_приложения>.heroku.com`.
9. Перейти по адресу `http://<имя_приложения>.herokuapp.com` и проверить работу всех функций приложения

Контрольная работа: Проект разработки и развертывания веб-приложения

1. Выбрать вариант для разработки базы данных
2. Скачать и запустить OpenServer Panel, запустить PHP MyAdmin, перейдя по адресу `http://localhost/openserver/phpmyadmin/`

3. Реализовать базу данных в PHP MyAdmin
4. Выполнить экспорт БД в файл при помощи вкладки Экспорт. Сохранить файл сценария.
5. Перейти во вкладку "Дизайнер". Построить и сохранить диаграмму данных.
6. Написать и отладить SQL-запросы к БД согласно своему варианту. Получить результат запроса на репрезентативном наборе данных.
7. Для получения зачета по контрольной работе необходимо отправить задание на проверку, прикрепив к нему файл сценария БД (*.sql) и скриншот схемы БД, построенной в дизайнера, тексты SQL-запросов и скриншоты с результатами их выполнения.
8. Разработать web-приложение для работы с базой данных MySQL согласно варианту задания.
9. Действующий пример приложения размещен в локальной сети по адресу <http://acs-srv2.surgu.ru/enterprise> (файловый доступ \\acs-srv2\www\enterprise), а также в репозитории BitBucket <https://2repby@bitbucket.org/2repby/enterprise.git>
10. Модифицированный сценарий enterprise.sql для создания объектов БД находится в папке с приложением.

Критерии оценивания лабораторных работ

1. Оценка «отлично»: правильно выполнены все задания в соответствии с требованиями, правильно выполнены дополнительные задания, своевременно предоставлен отчет о выполнении работы.
2. Оценка «хорошо»: правильно выполнены все задания в основной части, дополнительные задания выполнены не в полном объеме, предоставлен отчет о выполнении работы, либо в случае несвоевременного предоставления отчета или с наличием несущественных ошибок в выполнении лабораторных заданиях.
3. Оценка «удовлетворительно»: выполнены не все, но более 50% заданий лабораторной работы, дополнительные задания не выполнены, несвоеременно предоставлен отчет о выполнении работы.
4. Оценка «неудовлетворительно»: выполнено менее 50% лабораторной работы, не выполнены дополнительные задания, отчет о выполнении работы не предоставлен.

Критерии оценивания контрольной работы

1. Оценка «отлично»: правильно выполнены все задания в соответствии с требованиями, правильно выполнены дополнительные задания, своевременно предоставлен отчет о выполнении работы.
2. Оценка «хорошо»: правильно выполнены все задания в основной части, дополнительные задания выполнены не в полном объеме, предоставлен отчет о выполнении работы, либо в случае несвоевременного предоставления отчета или с наличием несущественных ошибок в выполнении контрольных заданий.
3. Оценка «удовлетворительно»: выполнены не все, но более 50% заданий контрольной работы, дополнительные задания не выполнены, несвоеременно предоставлен отчет о выполнении работы.
4. Оценка «неудовлетворительно»: выполнено менее 50% контрольной работы, не выполнены дополнительные задания, отчет о выполнении работы не предоставлен.

Этап: проведение промежуточной аттестации по дисциплине (ЗАЧЕТ)

Вопросы для проведения зачета с оценкой:

1. История появления сети Интернет. Основные события и факты.
2. Принцип многоуровневой организации передачи данных в Интернет.
3. Инкапсуляция пакетов и модель взаимодействия открытых систем.

4. Понятие протокола. Основные протоколы стека TCP/IP.
5. Назначение и основные характеристики протокола HTTP.
6. Структура HTML-документа, основные HTML-элементы форматирования текста в HTML.
7. Структура HTML-документа, основные HTML-элементы для подготовки списков и таблиц.
8. Назначение и синтаксис CSS. Виды CSS-селекторов. Способы подключения CSS-стилей к документу.
9. Назначение HTML-форм. Элементы HTML для определения форм
10. Объектная модель документа DOM. Работа с HTML-страницей на языке JavaScript.
11. Объектная модель браузера BOM. Работа с окнами и фреймами на языке JavaScript.
12. Работа с CSS-свойствами элементов HTML на языке JavaScript.
13. Обработка форм на стороне сервера на языке PHP. Передача параметров.
14. Способы аутентификации HTTP. Пример реализации basic-аутентификации на языке PHP.
15. Механизм cookies. Работа с cookies на языке PHP.
16. Сессии PHP. Преимущества и примеры работы с сессиями.